

CESM Tutorial

Adapted from NCAR CESM Tutorial
Materials
(Thank you NCAR CESM Team)

Outline

- **What is CESM?**
- **Release Homepage on Web**
- **Software & Hardware Requirements**
- **Basic Work Flow**
- **One-Time Setup**
 - **Registration and Source Code Download**
 - **Create an Input Data Root Directory**
 - **Porting**
- **Creating & Running a Case**
 - **Create a New Case**
 - **Configure the Case**
 - **Build the Executable**
 - **Initial Run and Outuput Data**
 - **Continuation Runs**
- **Getting More Help**
- **Appendix**



CESM1 Release Web Page

<http://www.cesm.ucar.edu/models/cesm1.0/>

Notable Improvements

Data, Diagnostics, and Post-Processing Tools

User's Guide

Component Model Documentation

External Libraries

Input Data

Timing Table

The screenshot shows the CESM1 Release Web Page with several sections:

- ABOUT CESM 1.0:** Describes CESM as a coupled climate model for simulating the earth's climate system.
- MODEL OUTPUT DATA AND DIAGNOSTICS:** Includes links to Model Output Diagnostic Tools, Model Output Data (ESD), and Post Processing Utilities.
- MODEL DOCUMENTATION:** Features a "How To Guide" and sections for Atmosphere Models, Ocean Models, Land Models, and Sea Ice Models.
- EXTERNAL LIBRARY DOCUMENTATION:** Lists Parallel I/O Library (PIO), Model Coupling Toolkit (MCT), and Earth System Modeling Framework (ESMF).
- MODEL INPUT DATA:** Notes that input data is available from a public WebDAV input data repository and provides instructions for checking out data.
- PERFORMANCE AND LOAD BALANCING DATA:** Provides performance data that will continue to evolve.
- CESM PROJECT:** Describes the CESM as a full-coupled, global climate model.
- CESM is sponsored by the National Science Foundation (NSF) and the U.S. Department of Energy (DOE). Administration of CESM is maintained by the Climate and Global Dynamics Division (CGD) at the National Center for Atmospheric Research (NCAR).**
- MODEL SOURCE CODE:** Contains links to Copyright and Terms of Use, Acquiring the Code, Version Summaries and Known Problems, and Reporting a Problem.

Background and Sponsors

Copyright and Terms of Use

How to Acquire the Code

Version Summaries

Known Problems and Reporting Problems

Software & Hardware Requirements

- Subversion client (version 1.4.2 or greater)
- Fortran and C compilers (recommend pgi, intel, or ibm xlf compilers)
- Netcdf library (recommend netcdf3.6.x)
- MPI (MPI1 is adequate, openmpi or mpich seem to work on linux clusters)
- CESM currently runs on “out of the box” today on the following machines
 - [bluefire](#) – NCAR IBM AIX
 - [jaguar](#) and [jaguarpf](#) – ORNL Cray XT4 and XT5
 - [franklin](#) and [hopper](#) – NERSC Cray XT4 and XT5
 - [kraken](#) – NICS Cray XT5
 - [intrepid](#) – ANL IBM Bluegene/P
 - [edinburgh](#) – NCAR linux cluster
 - [hadley](#) – LBL linux cluster
 - [midnight](#) – ARSC Sun cluster
 - [brutus](#) – ETH linux cluster
 - with pgi/openmpi, pgi/mpich, intel/openmpi, or intel/mpich
 - [and a few others](#)

Scripts have been created to run CESM on LANL machines ([lobo](#), [mapache](#), [conejo](#)).

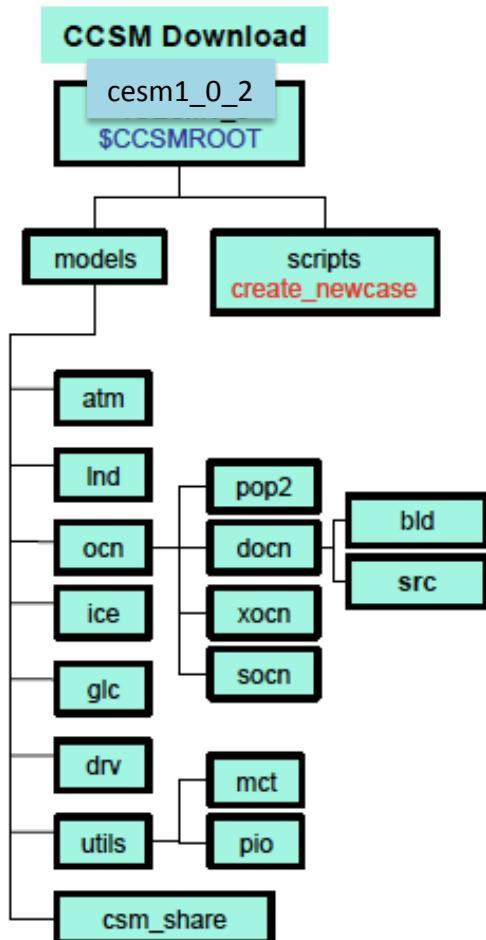
Basic Work Flow

- **One-Time Setup Steps**
 - (A) Registration and Download
 - (B) Create an Input Data Root Directory
 - (E) Porting
- **Creating & Running a Case**
 - (1) Create a New Case
 - (2) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

(A) CESM code

- Register for access to the CESM repo at
http://www.cesm.ucar.edu/models/cesm1.0/register/register_cesm1.0.cgi
- Check out CESM:
 - list available versions first – there may be newer releases
 - svn –username guestuser list https://svn-ccsm-release.cgd.ucar.edu/model_versions
 - svn –username guestuser co https://svn-ccsm-release.cgd.ucar.edu/model_versions/cesm1_0_1 cesm1_0_1
- Keep your own copy of the code (e.g., in /usr/projects/...)
- You can then copy the machine-specific scripts into your copy of CESM

(A) Overview of Directories (initial model download)



Next Step In the Basic Work Flow

- **One-Time Setup Steps**
 - (A)Registration and Download
 - (B) Create an Input Data Root Directory**
 - (C) Porting
- **Creating & Running a Case**
 - (1) Create a New Case
 - (2) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

(B) Input Data Directory

/usr/projects/cesm/input_data

All users in the “cesm” group (that probably means you) have permission to access and write to the input_data directory.

When running a case, CESM will check to see if the appropriate data are already in the input data directory (script “check_input_data”). This script downloads only the data needed for a particular run via svn.

By having everyone access this directory, the data directory will be “built up” over time, and the input data needed for any case will be available to users.

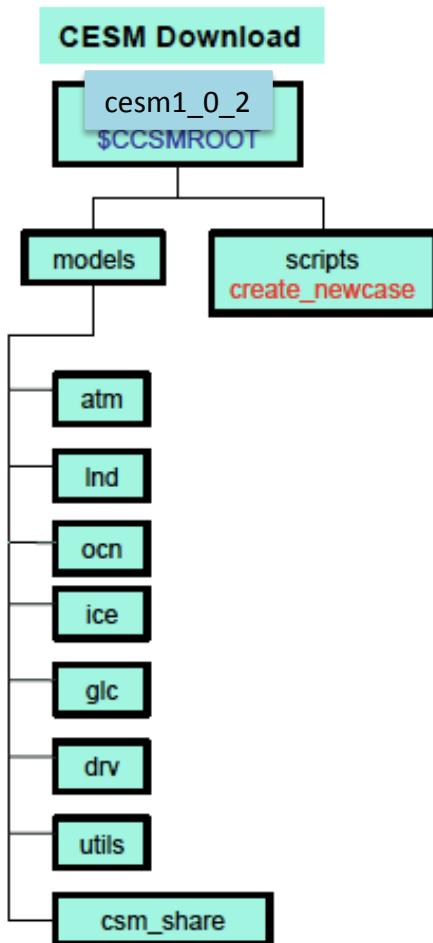
DON’T use the cesm directory as a personal directory (i.e., set up a workspace). DON’T modify or delete any job scripts or software. If you want to modify something, copy it to your own workspace and then tinker with it. This is a shared resource.



(B) Overview of Directories (+ inputdata directory)

INPUTDATA Directory

/usr/projects/cesm/input_data
\$DIN_LOC_ROOT



Next Step In the Basic Work Flow

- **One-Time Setup Steps**
 - (A)Registration and Download
 - (B)Create an Input Data Root Directory
 - (C) Porting**
- **Creating & Running a Case**
 - (1) Create a New Case
 - (2) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

(C) Porting

- Porting has already been done for mapache with pgi/9.0-3 with open_mpi/1.4.3.
 - The netcdf library is located in /usr/projects/cesm/software/conejo/netcdf.
 - This is hard-coded in Macros.mapache but can be changed if you have your own library.
 - Note that software that works on conejo works on mapache and vice versa.
- **/usr/projects/cesm/scripts/mapache/pgi contains four machine-specific files that you will have to copy into your version of CESM:**
env_machopts.mapache -> specifies compiler, mpi modules
mkbatch.mapache -> will make a Moab batch job for mapache
config_machines.xml -> defines cesm env variables (like EXEROOT="/scratch1/\$CCSMUSER/CESM/\$CASE" -> where cesm will run)
Macros.mapache -> build info (compiler options, libraries)

(C) “Porting” for Mapache:

Copy these scripts into your cesm directory (where you downloaded the code). You need to do this only once unless you get a new version of the code, or if the scripts change.

```
#cd to your cesm directory
#example directory used here
cd /usr/projects/climate/username/cesm1_0_2

#create a backup of config_machines.xml
cp scripts/ccsm_utils/Machines/config_machines.xml
    scripts/ccsm_utils/Machines/config_machines.xml.bak

#copy mapache-specific scripts
cp /usr/projects/cesm/scripts/mapache/pgi/* scripts/ccsm_utils/Machines/
```

Mapache Quickstart

- We will do the following:
 - Run an initial 5-day test (for quick output)
 - Change the length of the run to a full month (to see a full month history file) and re-do the simulation.
 - On your own: perform two 3-year length runs as continuation runs.

Mapache Quickstart

```
# set up the cesm environment on mapache
source /usr/projects/cesm/scripts/mapache/pgi/cesm_env_script

# go to root directory of cesm code and cd to scripts directory
cd cesm1_0_2/scripts

# (1) create a new case
./create_newcase -case mycase.1850 -res T31_g37 -compset B_1850 -mach mapache

# (2) configure the case
cd mycase.1850
./configure --case

# (3) build the case
./mycase.1850.mapache.build

# (4) submit the case
./mycase.1850.mapache.submit

#check job status
showq -w user=username
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# (5) when the run finishes, change the length from a continuation run to a month-long
#     run and continue the simulation, then check job status as above (from mycase dir)
xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE
./mycase.1850.mapache.submit

# (6) view job output after job has completed
source Tools/ccsm_getenv

cd $DOUT_S_ROOT/atm/hist
ncview mycase.1850.cam2.h1.0001-01-01-00000.nc
```

Next Step In the Basic Work Flow

- **One-Time Setup Steps**
 - (A) Registration and Download
 - (B) Create an Input Data Root Directory
 - (C) Porting
- **Creating & Running a Case**
 - (1) Create a New Case**
 - (2) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Runs

Mapache Quickstart

```
# set up the cesm environment on mapache
source /usr/projects/cesm/scripts/mapache/pgi/cesm_env_script

# go to root directory of cesm code and cd to scripts directory
cd cesm1_0_2/scripts

# (1) create a new case
./create_newcase -case mycase.1850 -res T31_g37 -compset B_1850 -mach mapache

# (2) configure the case
cd mycase.1850
./configure --case

# (3) build the case
./mycase.1850.mapache.build

# (4) submit the case
./mycase.1850.mapache.submit

#check job status
showq -w user=username
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# (5) when the run finishes, change the length from a continuation run to a month-long
#     run and continue the simulation, then check job status as above (from mycase dir)
xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE
./mycase.1850.mapache.submit

# (6) view job output after job has completed
source Tools/ccsm_getenv

cd $DOUT_S_ROOT/atm/hist
ncview mycase.1850.cam2.h1.0001-01-01-00000.nc
```



(1) Create a New Case

- Top of scripts directory contains key scripts
- First step in setting up a model run is running the script `create_newcase`
- Go to the scripts directory: .../CESM1_0/scripts/
 - `create_newcase` is the tool that generates a new case
- Scripts are a combination of csh, perl, sh, and xml.

```
CESM1_0/scripts>ls -l
total 400
-rw-r--r--    1 userx  ncar          18596 May 12 11:33 ChangeLog
-rw-r--r--    1 userx  ncar           168 May 12 11:33 README
-rw-r--r--    1 userx  ncar           103 May 12 11:33 SVN_EXTERNAL_DIRECTORIES
drwxr-xr-x   10 userx  ncar          8192 May 12 11:33 ccsm_utils
-rwxr-xr-x    1 userx  ncar          19039 May 12 11:33 create_clone
-rwxr-xr-x    1 userx  ncar          52338 May 12 11:33 create_newcase ← create_newcase
-rwxr-xr-x    1 userx  ncar          18253 May 12 11:33 create_test
-rwxr-xr-x    1 userx  ncar           9643 May 12 11:33 create_test_suite
drwxr-xr-x    3 userx  ncar          8192 May 12 11:33 doc
-rwxr-xr-x    1 userx  ncar          1255 May 12 11:33 link_dirtree
-rw-r--r--    1 userx  ncar           295 May 12 11:33 sample_compset_file.xml
-rw-r--r--    1 userx  ncar           851 May 12 11:33 sample_pes_file.xml
```



(1) About `create_newcase`

- `create_newcase` has many command line options - most are rarely used
- `create_newcase -help` lists all the available options
- Most often only four options are used: case, compset, res, and mach

```
CESM1_0/scripts>./create_newcase -help
SYNOPSIS
    create_newcase [options]
OPTIONS
User supplied values are denoted in angle brackets (<>). Any value that contains
white-space must be quoted. Long option names may be supplied with either single
or double leading dashes. A consequence of this is that single letter options may
NOT be bundled.

    -case <name>          Specifies the case name (required).
    -compset <name>        Specify a CESM compset (required).
    -res <name>            Specify a CCSM grid resolution (required).
    -mach <name>           Specify a CESM machine (required).
    -pecount <name>         Value of S,M,L,X1,X2 (optional). (default is M).
    -pes_file <name>        Full pathname of pes setup file to use (will overwrite default settings) (optional).
    -compset_file <name>    Full pathname of compset setup file to use. (optional)

    -help [or -h]           Print usage to STDOUT (optional).
    -list                  Only list valid values for compset, grid settings and machines (optional).
    -silent [or -s]         Turns on silent mode - only fatal messages issued (optional).
    -verbose [or -v]        Turn on verbose echoing of settings made by create_newcase (optional).
    -xmlmode <name>         Sets format of xml files; normal or expert (optional). (default is normal)

The following arguments are required for a generic machine. Otherwise, they will be ignored.

    -scratchroot <name>      CCSM executable directory (EXEROOT will be scratchroot/CASE) (char)
    -din_loc_root_csmdata <name>  csm input data root directory (char)
    -max_tasks_per_node <value>   maximum mpi tasks per machine node (integer)

The following two arguments turn on single point mode.
If one is given -- both MUST be given.

    -pts_lat <value>          Latitude of single point to operate on (optional)
    -pts_lon <value>           Longitude of single point to operate on (optional)
```



(1) `create_newcase` -- Four Required Arguments

```
./create_newcase -case ~/cases/mycase1 -res f19_g16 -compset B_1850 -mach bluefire
```

- “**case**” is the name and location of the case being created
 - `~/cases/mycase1`
- “**res**” specifies the model resolutions (or grid)
 - Format is [atm/lnd grid]_[ocn/ice grid], eg., `f19_g16` is 1.9x2.5 atm/lnd + gx1v6 ocn/ice
 - Most often the atm & lnd share the same grid, and the ice & ocn share the same grid
 - Equivalent short and long names (`f19_g16 == 1.9x2.5_gx1v6`)
- “**compset** specifies the “component set”
 - component set specifies component models and physics options for those models
 - Eg. “B” compsets use all active models (CAM,CLM,CICE,POP2)
 - Eg. “F” compsets use CAM,CLM, CICE(prescribed-thermo), DOCN(prescribed-SST)
 - Equivalent short and long names (`B1850CN == B_1850_CN`)
- “**mach**” specifies the machine that will be used.
 - “supported” machines tested regularly, eg. `bluefire, jaguar, franklin, intrepid`
 - “prototype” machines are not tested regularly, eg. `prototype_frost`
 - “generic machines” provide a starting point for porting, eg. “`generic_ibm`
- **`create_newcase -list`**
 - lists all the valid choices for these command line options (see next slide)
- **values are set on the command line are “locked down” in case directory**
 - file `env_case.xml` contains all “locked down” variables when `create_newcase` was run

(1) Valid Values for res, compset, and mach

```
CESM1_0/scripts>./create_newcase -list

RESOLUTIONS: name (shortname)
 0.9x1.25_0.9x1.25 (f09_f09)
 0.9x1.25_gxlv6 (f09_g16)
 1.9x2.5_1.9x2.5 (f19_f19)
 1.9x2.5_gxlv6 (f19_g16)
 4x5_gx3v7 (f45_g37)
 T31_gx3v7 (T31_g37)
 ne30np4_1.9x2.5_gxlv6 (ne30_f19_g16)

COMPSETS: name (shortname): description (status)
 A_PRESENT_DAY (A)
   Description: All data model
 B_2000 (B)
   Description: All active components, present day
 B_1850 (B1850)
   Description: All active components, pre-industrial
 B_1850_CN (B1850CN)
   Description: all active components, pre-industrial, with CN (Carbon Nitrogen) in CLM
 F_AMIP (FAMIP)
   Description: Default resolution independent AMIP is INVALID
 F_2000_CN (FCN)
   Description: Stand-alone cam default, prescribed ocn/ice with CN
 G_NORMAL_YEAR (G)
   Description: Coupled ocean ice with COREv2 normal year forcing
 I_2000 (I)
   Description: Active land model with QIAN atm input data for 2003 and Satellite phenology (SP), CO2 level
                 and Aerosol deposition for 2000
 I_1850 (I1850)
   Description: Active land model with QIAN atm input data for 1948 to 1972 and Satellite phenology (SP), CO2
                 level and Aerosol deposition for 1850

MACHINES: name (description)
 bluefire (NCAR IBM p6, os is AIX, 32 pes/node, batch system is LSF)
 franklin (NERSC XT4, os is CNL, 4 pes/node, batch system is PBS)
 intrepid (ANL IBM BG/P, os is BGP, 4 pes/node, batch system is cobalt)
 jaguar (ORNL XT4, os is CNL, 4 pes/node, batch system is PBS)
 jaguarpf (ORNL XT5, os is CNL, 12 pes/node, batch system is PBS)
 prototype_ranger (TACC Linux Cluster, Linux (pgi), 1 pes/node, batch system is SGE)
 generic_linux_pgi (generic linux (pgi), os is Linux, batch system is PBS, user-defined)
 generic_linux_intel (generic linux (intel), os is Linux, batch system is PBS, user-defined)
```

```
./create_newcase -case mycase1 -res f19_g16 -compset B_1850 -mach mapache
```

CESM1.0 README

For both a quick start as well as a detailed summary of creating and running a CESM model case, see the CESM1.0 User's Guide at
<http://www.cesm.ucar.edu/models/cesm1.0>

IMPORTANT INFORMATION ABOUT SCIENTIFIC VALIDATION

CESM1.0 has the flexibility to configure cases with many different combinations of component models, grids, and model settings, but this version of CESM has only been validated scientifically for the following fully active configurations:

```
1.9x2.5_gx1v6 B_1850_CN
1.9x2.5_gx1v6 B_1850_RAMP_C02_CN
1.9x2.5_gx1v6 B_1850_2000_CN

1.9x2.5_gx1v6 B_1850_CAM5

0.9x1.25_gx1v6 B_1850_CN
0.9x1.25_gx1v6 B_1850_RAMP_C02_CN
0.9x1.25_gx1v6 B_1850_2000_CN

0.9x1.25_gx1v6 B_1850_BGC-BPRP
0.9x1.25_gx1v6 B_1850_BGC-BORD
0.9x1.25_gx1v6 B_1850_2000_BGC-BPRP
0.9x1.25_gx1v6 B_1850_2000_BGC-BORD

0.9x1.25_gx1v6 B_1850_CN_CHEM
0.9x1.25_gx1v6 B_1850_2000_CN_CHEM

1.9x2.5_gx1v6 B_1850_WACCM_CN
1.9x2.5_gx1v6 B_1850_2000_WACCM_CN

T31_gx3v7 B_1850_CN
```

If the user is interested in running a "stand-alone" component configuration, the following model configurations have been validated scientifically and have associated diagnostic output as part of the release:

```
1.9x2.5_1.9x2.5 F_2000_WACCM
1.9x2.5_1.9x2.5 F_AMIP_CAM5
1.9x2.5_1.9x2.5 F_AMIP_CN
0.9x1.25_0.9x1.25 F_AMIP_CN

0.9x1.25_gx1v6 I_2000
0.9x1.25_gx1v6 I_2000_CN

T62_gx1v6 C_NORMAL_YEAR
```

For more information regarding alternative component configurations, please refer to the individual component web pages at
<http://www.cesm.ucar.edu/models/cesm1.0>

```
*****
Component set      : B_1850 (B1850)
Desc       : All active components, pre-industrial
*****
```

```
Creating /usr/projects/climate/rauscher/cesm1_0_2/scripts/mycase1
```

```
Locking file /usr/projects/climate/rauscher/cesm1_0_2/scripts/mycase1/env_case.xml
Successfully created the case for mapache
```

Result of running ./create_newcase

Warning message about
scientific validation (more
about that from Cecile)

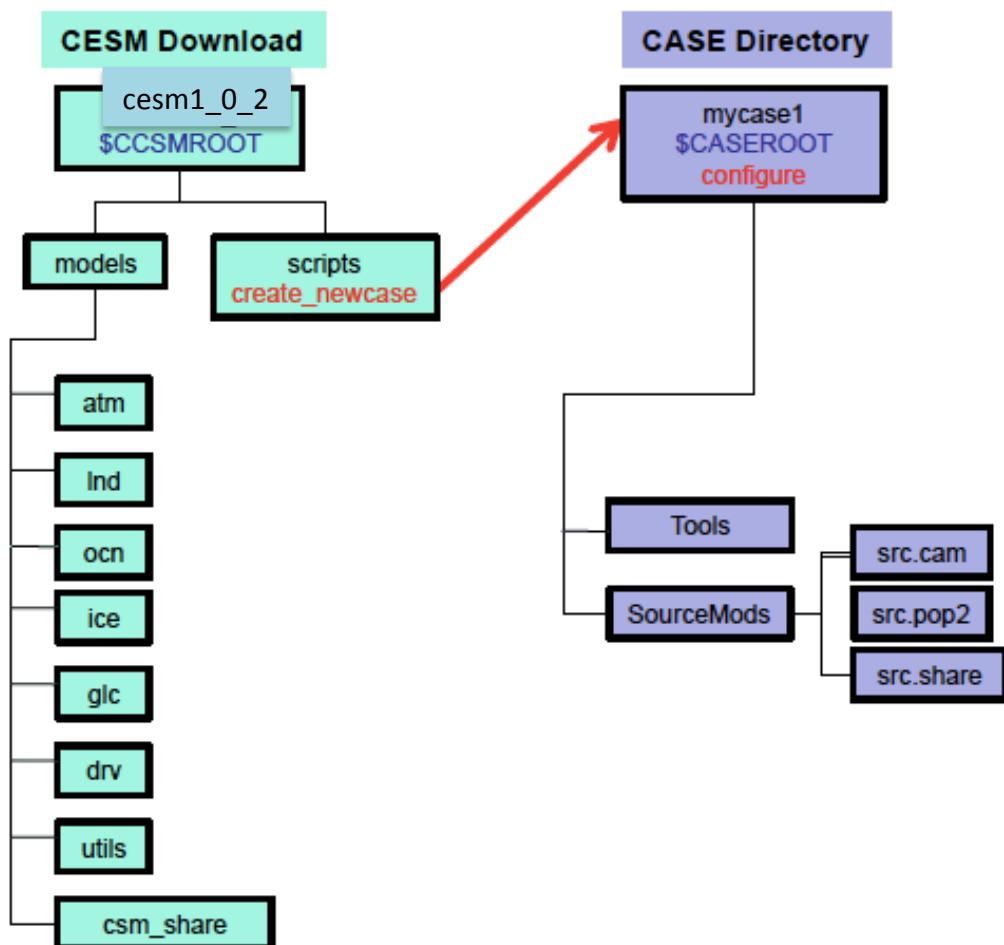
Case location

Success!



(1) Overview of Directories (+ create_newcase)

INPUTDATA Directory
/usr/projects/cesm/input_data
\$DIN_LOC_ROOT



(1) Case Directory After Running `create_newcase`

- **SourceMods** is a directory where case specific code modification can be placed
- **configure** is the script used in the next step, step (2)
- **env_*.xml** contain environment variables associate with the case (more on this later)
- **xmlchange** is a script that changes env variable values through a command line interface

```
CESM1_0/scripts> cd ~/cases/mycasel
cases/mycasel>ls -l
total 64
drwxr-xr-x  2 userx  ncar          8192 May 13 14:32 LockedFiles
-rw-r--r--  1 userx  ncar         10687 May 13 14:32 Macros.bluefire
drwxr-xr-x  2 userx  ncar          8192 May 13 14:32 README
-rw-r--r--  1 userx  ncar           66 May 13 14:32 README.case
drwxr-xr-x  9 userx  ncar          8192 May 13 14:32 SourceMods
drwxr-xr-x  4 userx  ncar          8192 May 13 14:32 Tools
-rwxr-xr-x  1 userx  ncar         9330 May 12 11:33 check_input_data
-rwxr-xr-x  1 userx  ncar        10092 May 12 11:33 configure
-rwxr-xr-x  1 userx  ncar         3085 May 12 11:33 create_production_test
-rw-r--r--  1 userx  ncar         4433 May 13 14:32 env_build.xml
-rw-r--r--  1 userx  ncar         5635 May 13 14:32 env_case.xml
-rw-r--r--  1 userx  ncar         7029 May 13 14:32 env_conf.xml
-rw-r--r--  1 userx  ncar         5915 May 13 14:32 env_mach_pes.xml
-rw-r--r--  1 userx  ncar         2199 May 13 14:32 env_mach_specific
-rw-r--r--  1 userx  ncar        10466 May 13 14:32 env_run.xml
-rwxr-xr-x  1 userx  ncar        10388 May 12 11:33 xmlchange
```

SourceMods

configure

env files

xmlchange



(*) About .xml Files: Format & Variables

- Contain variables used by scripts -- some can be changed by the user
- Here's a snippet of the env_run.xml file

```
<!--sets the run length in conjunction with STOP_N and STOP_DATE, valid values: none,never,nst  
eps,nstep,nseconds,nsecond,nminutes,nminute,nhours,nhour,ndays,nday,nmonths,nmonth,nyears,nyea  
r,date,ifdays0,end (char) " -->  
<entry id="STOP_OPTION"    value="ndays"  />  
  
<!--sets the run length in conjunction with STOP_OPTION and STOP_DATE (integer) " -->  
<entry id="STOP_N"      value="5"     />  
  
<!--logical to turn on short term archiving, valid values: TRUE,FALSE (logical) " -->  
<entry id="DOUT_S"      value="TRUE"   />  
  
<!--local short term archiving root directory (char) " -->  
<entry id="DOUT_S_ROOT"  value="/ptmp/$CCSMUSER/archive/$CASE"  />
```

- “**id**” is the variable name
- “**value**” is the variables name’s setting
- <!-- text --> is a comment in xml (most variables have a description above the entry)
 - “**(type)**” is the type of the variable
 - “**valid values**” indicates the full set of allowable settings
 - the scripts will let you know if you try to set a variable to an invalid value
 - many values do not have valid values defined, that means there are no constraints
- To modify a variable in an xml file
 - use **xmlchange** to modify env variable settings
 - > **xmlchange -help**
 - > **xmlchange -file env_run.xml -id STOP_N -val 20**
 - edit env_*.xml file manually -- but be careful about introducing formatting errors
- To see how xml variables translate to your local environment
 - > **source ./Tools/ccsm_getenv**

(*) About .xml Files: How They Change the Build and Run

- Defaults are generally reasonable
- **env_case.xml** is set by `create_newcase` and cannot be modified
- **env_conf.xml** variables specify various component information
 - Most often this file should not be modified
 - **RUN_TYPE, RUN_STARTDATE, RUN_REFCASE, RUN_REFDATE** – defines initial conditions
 - Can change the physics of a model – be very careful about this
- **env_mach_pes.xml** variables specify the layout of components on hardware processors
 - Is used to tune the performance of the model - **scientific results do not depend on component/processor layout**
 - **NTASKS_*** - number of mpi tasks assigned to the component
 - **NTHRDS_*** - number of openmp threads per mpi task for the component
 - **ROOTPE_*** - global mpi task rank of the component root mpi task
- **env_build.xml** variables specify some build information
 - Most often this file should not be modified
- **Macros.*** specifies the compilation variables used in the Makefile
 - Most often this file should not be modified
- **env_mach_specific**
 - Sets modules and paths to libraries (e.g. MPI)
 - Can change compiler options, libraries, etc.
 - Part of porting is to set variables here
- **env_run.xml** variables specify run time information
 - Most often this file *will be modified*
 - **STOP_OPTION, STOP_N, REST_OPTION, REST_N**



```
<entry id="NTASKS_ATM" value="64" />
<entry id="NTHRDS_ATM" value="1" />
<entry id="ROOTPE_ATM" value="0" />

<entry id="NTASKS_LND" value="64" />
<entry id="NTHRDS_LND" value="1" />
<entry id="ROOTPE_LND" value="0" />

<entry id="NTASKS_ICE" value="64" />
<entry id="NTHRDS_ICE" value="1" />
<entry id="ROOTPE_ICE" value="0" />

<entry id="NTASKS_OCN" value="64" />
<entry id="NTHRDS_OCN" value="1" />
<entry id="ROOTPE_OCN" value="0" />

<entry id="NTASKS_CPL" value="64" />
<entry id="NTHRDS_CPL" value="1" />
<entry id="ROOTPE_CPL" value="0" />
```

Next Step In the Basic Work Flow

- **One-Time Setup Steps**
 - (A)Registration and Download
 - (B)Create an Input Data Root Directory
 - (C) Porting
- **Creating & Running a Case**
 - (1)Create a New Case
 - (2) Configure the Case**
 - (3) Build the Executable
 - (4)Running the Model: Initial Run and Output Data
 - (5)Running the Model: Continuation Runs

Mapache Quickstart

```
# set up the cesm environment on mapache
source /usr/projects/cesm/scripts/mapache/pgi/cesm_env_script

# go to root directory of cesm code and cd to scripts directory
cd cesm1_0_2/scripts

# (1) create a new case
./create_newcase -case mycase.1850 -res T31_g37 -compset B_1850 -mach mapache

# (2) configure the case
cd mycase.1850
./configure --case

# (3) build the case
./mycase.1850.mapache.build

# (4) submit the case
./mycase.1850.mapache.submit

#check job status
showq -w user=username
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# (5) when the run finishes, change the length from a continuation run to a month-long
# run and continue the simulation, then check job status as above (from mycase dir)
xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE
./mycase.1850.mapache.submit

# (6) view job output after job has completed
source Tools/ccsm_getenv

cd $DOUT_S_ROOT/atm/hist
ncview mycase.1850.cam2.h1.0001-01-01-00000.nc
```



(2) Configure the Case

- Step (2) is to “configure” the case using the **configure** script
- Run **configure --case**
 - Can modify **env_conf.xml** and **env_mach_pes.xml** *before running configure*, but not after (unless invoke **configure** with a clean option)
 - Most often there is no need to modify **env_conf.xml** or **env_mach_pes.xml**
- Generates
 - **Buildconf/** directory with **buildnml**, **buildexe**, and **input_data_list** files
 - **case *.build** and ***.run** scripts
- Locks **env_conf.xml** and **env_mach_pes.xml**

```
CESM1_0/scripts> cd ~/cases/mycasel
cases/mycasel>ls -l
total 64
drwxr-xr-x  2 userx  ncar          8192 May 13 14:32 LockedFiles
-rw-r--r--  1 userx  ncar         10687 May 13 14:32 Macros.bluefire
drwxr-xr-x  2 userx  ncar          8192 May 13 14:32 README
-rw-r--r--  1 userx  ncar           66 May 13 14:32 README.case
drwxr-xr-x  9 userx  ncar          8192 May 13 14:32 SourceMods
drwxr-xr-x  4 userx  ncar          8192 May 13 14:32 Tools
-rwxr-xr-x  1 userx  ncar         9330 May 12 11:33 check_input_data
-rwxr-xr-x  1 userx  ncar        10092 May 12 11:33 configure ←
-rwxr-xr-x  1 userx  ncar         3085 May 12 11:33 create_production_test
-rw-r--r--  1 userx  ncar         4433 May 13 14:32 env_build.xml
-rw-r--r--  1 userx  ncar         5635 May 13 14:32 env_case.xml
-rw-r--r--  1 userx  ncar         7029 May 13 14:32 env_conf.xml
-rw-r--r--  1 userx  ncar         5915 May 13 14:32 env_mach_pes.xml ←
-rwxr-xr-x  1 userx  ncar         2199 May 13 14:32 env_mach_specific
-rw-r--r--  1 userx  ncar        10466 May 13 14:32 env_run.xml
-rwxr-xr-x  1 userx  ncar        10388 May 12 11:33 xmlchange
```

configure

env_conf.xml
env_mach_pes.xml

(2) About configure

> **configure -help**

```
NAME
    configure - configures the model for a given resolution, component set
    and machine.

SYNOPSIS
    configure [-case] [-cleannamelist] [-cleanmach] [-cleanall]
```

- ***configure -case***
 - Generates Buildconf directory and buildnml, buildexe, and input_data_list files
 - Generates the case .build and .run scripts
 - Locks env_conf.xml and env_mach_pes.xml
- ***configure -cleanall***
 - Unlocks env_conf.xml and env_mach_pes.xml
 - “Backs up” Buildconf and run scripts
 - Modify env_conf.xml and env_mach_pes.xml and type configure -case again

Running ./configure -case

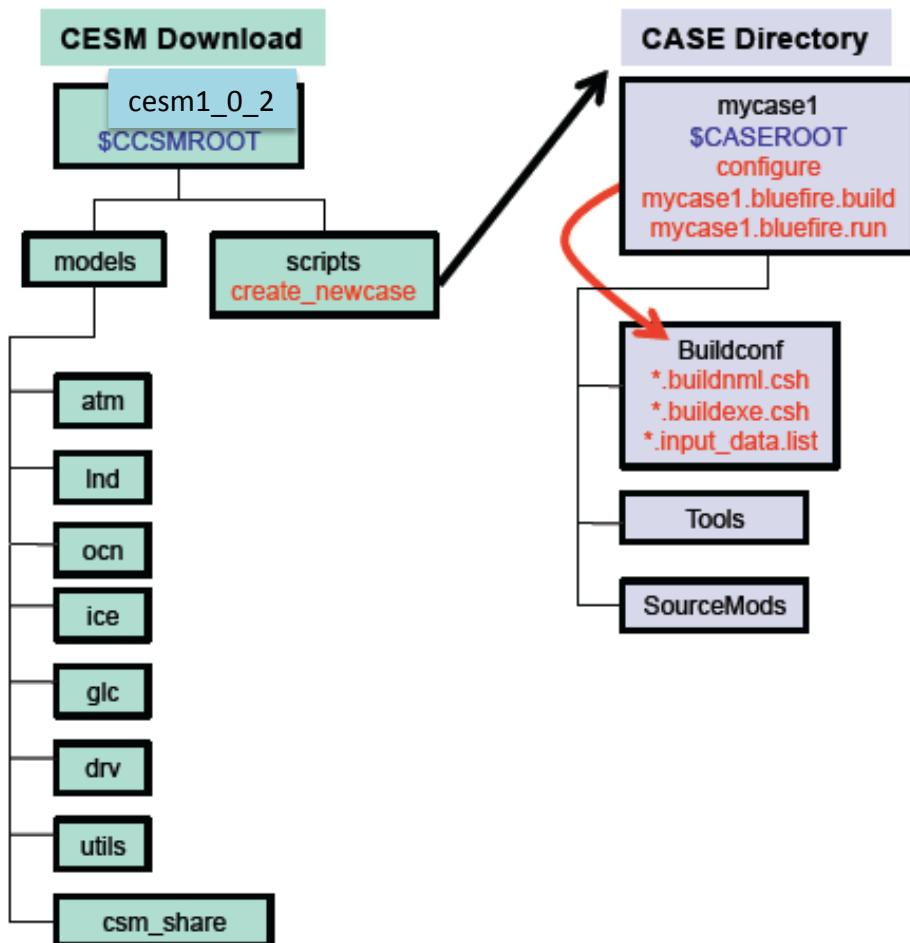
```
mp-fe2.lanl.gov-/usr/projects/climate/rauscher/cesm1_0_2/scripts/mycase1 % ./configure -case
Generating resolved namelist, prestage, and build scripts
configure done.
adding use_case 1850_control defaults for var sim_year with val 1850
adding use_case 1850_control defaults for var sim_year_range with val constant
adding use_case 1850_control defaults for var use_case_desc with val Conditions to simulate 1850 land-use
configure done.
Successfully generated resolved namelist, prestage, and build scripts
Locking file env_conf.xml
Generating clean_build script
Generating submit script
Generating build script
Generating run script
Locking file env_mach_pes.xml
Successfully configured the case for mapache
If an old build exists for this case, you might want to
  run the *.clean_build script before building
```



(2) Overview of Directories (+ configure)

INPUTDATA Directory

/usr/projects/cesm/input_data
\$DIN_LOC_ROOT



In our cases, “bluefire” is replaced with “mapache”



(2) Case Dir After Running `configure`

- `configure` adds the `Buildconf` directory and populates it
- `configure` generates `build`, `clean_build`, `run`, and `archive` scripts

```
cases/mycase1>ls -l
total 432
drwxr-xr-x  6 userx  ncar          8192 May 13 17:12 Buildconf ← Buildconf
drwxr-xr-x  2 userx  ncar          8192 May 13 17:12 LockedFiles
drwxr-xr-x  1 userx  ncar         10687 May 13 14:32 Macros.bluefire
drwxr-xr-x  2 userx  ncar          8192 May 13 14:32 README
drwxr-xr-x  1 userx  ncar           66 May 13 14:32 README.case
drwxr-xr-x  9 userx  ncar          8192 May 13 14:32 SourceMods
drwxr-xr-x  4 userx  ncar          8192 May 13 14:32 Tools
drwxr-xr-x  1 userx  ncar         9330 May 12 11:33 check_input_data
drwxr-xr-x  1 userx  ncar        10092 May 12 11:33 configure
drwxr-xr-x  1 userx  ncar         3085 May 12 11:33 create_production_test
drwxr-xr-x  1 userx  ncar         4454 May 13 17:12 env_build.xml
drwxr-xr-x  1 userx  ncar         5635 May 13 14:32 env_case.xml
drwxr-xr-x  1 userx  ncar         7029 May 13 14:32 env_conf.xml
drwxr-xr-x  1 userx  ncar          614 May 13 17:12 env_derived
drwxr-xr-x  1 userx  ncar         5916 May 13 17:12 env_mach_pes.xml
drwxr-xr-x  1 userx  ncar         2199 May 13 14:32 env_mach_specific
drwxr-xr-x  1 userx  ncar        10466 May 13 14:32 env_run.xml
drwxr-xr-x  1 userx  ncar          574 May 13 17:12 mycase1.bluefire.build ← new scripts
drwxr-xr-x  1 userx  ncar          836 May 13 17:12 mycase1.bluefire.clean_build
drwxr-xr-x  1 userx  ncar          802 May 13 17:12 mycase1.bluefire.l_archive
drwxr-xr-x  1 userx  ncar         3938 May 13 17:12 mycase1.bluefire.run
drwxr-xr-x  1 userx  ncar         10388 May 12 11:33 xmlchange
```

(2) Files in the Buildconf Directory (Created by `configure`)

- The `configure` script fills the Buildconf directory which contains
 - Component buildnml.csh scripts**
 - Component buildexe.csh scripts**
 - Component input_data.list**

```
cases/mycase1>ls -l Buildconf/
total 448
-rwxr-xr-x 1 userx ncar          850 May 13 17:12 cam.buildexe.csh
-rwxr-xr-x 1 userx ncar         3625 May 13 17:12 cam.buildnml.csh
-rwxr-xr-x 1 userx ncar         1508 May 13 17:12 cam.input_data_list
drwxr-xr-x 2 userx ncar        8192 May 13 17:12 camconf
-rwxr-xr-x 1 userx ncar          480 May 13 17:12 ccsm.buildexe.csh
-rwxr-xr-x 1 userx ncar         1414 May 13 17:12 cice.buildexe.csh
-rwxr-xr-x 1 userx ncar         3292 May 13 17:12 cice.buildnml.csh
-rwxr-xr-x 1 userx ncar          379 May 13 17:12 cice.input_data_list
drwxr-xr-x 2 userx ncar        8192 May 13 17:12 ciceconf
-rwxr-xr-x 1 userx ncar         1174 May 13 17:12 clm.buildexe.csh
-rwxr-xr-x 1 userx ncar         2269 May 13 17:12 clm.buildnml.csh
-rwxr-xr-x 1 userx ncar          702 May 13 17:12 clm.input_data_list
drwxr-xr-x 2 userx ncar        8192 May 13 17:12 clmconf
-rwxr-xr-x 1 userx ncar           42 May 13 17:12 cpl.buildexe.csh
-rwxr-xr-x 1 userx ncar        10507 May 13 17:12 cpl.buildnml.csh
-rwxr-xr-x 1 userx ncar         1665 May 13 17:12 csm_share.buildlib
-rwxr-xr-x 1 userx ncar         1965 May 13 17:12 mct.buildlib
-rwxr-xr-x 1 userx ncar         2412 May 13 17:12 pio.buildlib
-rwxr-xr-x 1 userx ncar         5546 May 13 17:12 pop2.buildexe.csh
-rwxr-xr-x 1 userx ncar        29056 May 13 17:12 pop2.buildnml.csh
-rwxr-xr-x 1 userx ncar         1012 May 13 17:12 pop2.input_data_list
drwxr-xr-x 2 userx ncar        8192 May 13 17:12 pop2doc
-rwxr-xr-x 1 userx ncar          588 May 13 17:12 sgfc.buildexe.csh
-rwxr-xr-x 1 userx ncar            78 May 13 17:12 sgfc.buildnml.csh
```

Next Step In the Basic Work Flow

- **One-Time Setup Steps**
 - (A)Registration and Download
 - (B)Create an Input Data Root Directory
 - (C) Porting
- **Creating & Running a Case**
 - (1)Create a New Case
 - (2)Configure the Case
 - (3) Build the Executable**
 - (4)Run the Model: Initial Run and Output Data Flow
 - (5)Run the Model: Continuation Runs

Mapache Quickstart

```
# set up the cesm environment on mapache
source /usr/projects/cesm/scripts/mapache/pgi/cesm_env_script

# go to root directory of cesm code and cd to scripts directory
cd cesm1_0_2/scripts

# (1) create a new case
./create_newcase -case mycase.1850 -res T31_g37 -compset B_1850 -mach mapache

# (2) configure the case
cd mycase.1850
./configure --case

# (3) build the case
./mycase.1850.mapache.build

# (4) submit the case
./mycase.1850.mapache.submit

#check job status
showq -w user=username
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# (5) when the run finishes, change the length from a continuation run to a month-long
#     run and continue the simulation, then check job status as above (from mycase dir)
xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE
./mycase.1850.mapache.submit

# (6) view job output after job has completed
source Tools/ccsm_getenv

cd $DOUT_S_ROOT/atm/hist
ncview mycase.1850.cam2.h1.0001-01-01-00000.nc
```

(3) Build the Model

- Step (3) is to “build” the model using the **.build** script
- Modifications before build
 - Can change values in env_build.xml *before* running *.build, but not after
 - Usually there is no need to edit env_build.xml
 - May want to introduce modified source code before building
- The **.build** script
 - checks for missing input data by running **check_input_data -check**
 - creates build/run directory that will contain executable code and model namelist files
 - locks env_build.xml
 - builds the individual component libraries and then final executable
- If any inputdata is missing...
 - The build will abort, but it will provide a list of missing files
 - You must run the script **check_input_data -export** to acquire missing data
 - **check_input_data -export** will use svn to put required data in the inputdata directory
 - You must re-run build script after running **check_input_data -export**

(3) The .build script

```
cases/mycasel>ls -l
total 432
drwxr-xr-X  6 userx  ncar          8192 May 13 17:12 Buildconf
drwxr-xr-X  2 userx  ncar          8192 May 13 17:12 LockedFiles
-rw-r--r--  1 userx  ncar        10687 May 13 14:32 Macros.bluefire
drwxr-xr-X  2 userx  ncar          8192 May 13 14:32 README
-rw-r--r--  1 userx  ncar           66 May 13 14:32 README.case
drwxr-xr-X  9 userx  ncar          8192 May 13 14:32 SourceMods
drwxr-xr-X  4 userx  ncar          8192 May 13 14:32 Tools
-rw-r--r--  1 userx  ncar         9330 May 12 11:33 check_input_data ←
drwxr-xr-X  1 userx  ncar        10092 May 12 11:33 configure
-rw-r--r--  1 userx  ncar         3085 May 12 11:33 create_production_test
-rw-r--r--  1 userx  ncar        4454 May 13 17:12 env_build.xml ←
drwxr-xr-X  1 userx  ncar         5635 May 13 14:32 env_case.xml
drwxr-xr-X  1 userx  ncar         7029 May 13 14:32 env_conf.xml
-rw-r--r--  1 userx  ncar          614 May 13 17:12 env_derived
drwxr-xr-X  1 userx  ncar         5916 May 13 17:12 env_mach_pes.xml
drwxr-xr-X  1 userx  ncar         2199 May 13 14:32 env_mach_specific
-rw-r--r--  1 userx  ncar        10466 May 13 14:32 env_run.xml
drwxr-xr-X  1 userx  ncar          574 May 13 17:12 mycasel.bluefire.build ←
drwxr-xr-X  1 userx  ncar          836 May 13 17:12 mycasel.bluefire.clean_build
drwxr-xr-X  1 userx  ncar          802 May 13 17:12 mycasel.bluefire.1_archive
drwxr-xr-X  1 userx  ncar         3938 May 13 17:12 mycasel.bluefire.run
drwxr-xr-X  1 userx  ncar        10388 May 12 11:33 xmlchange
```

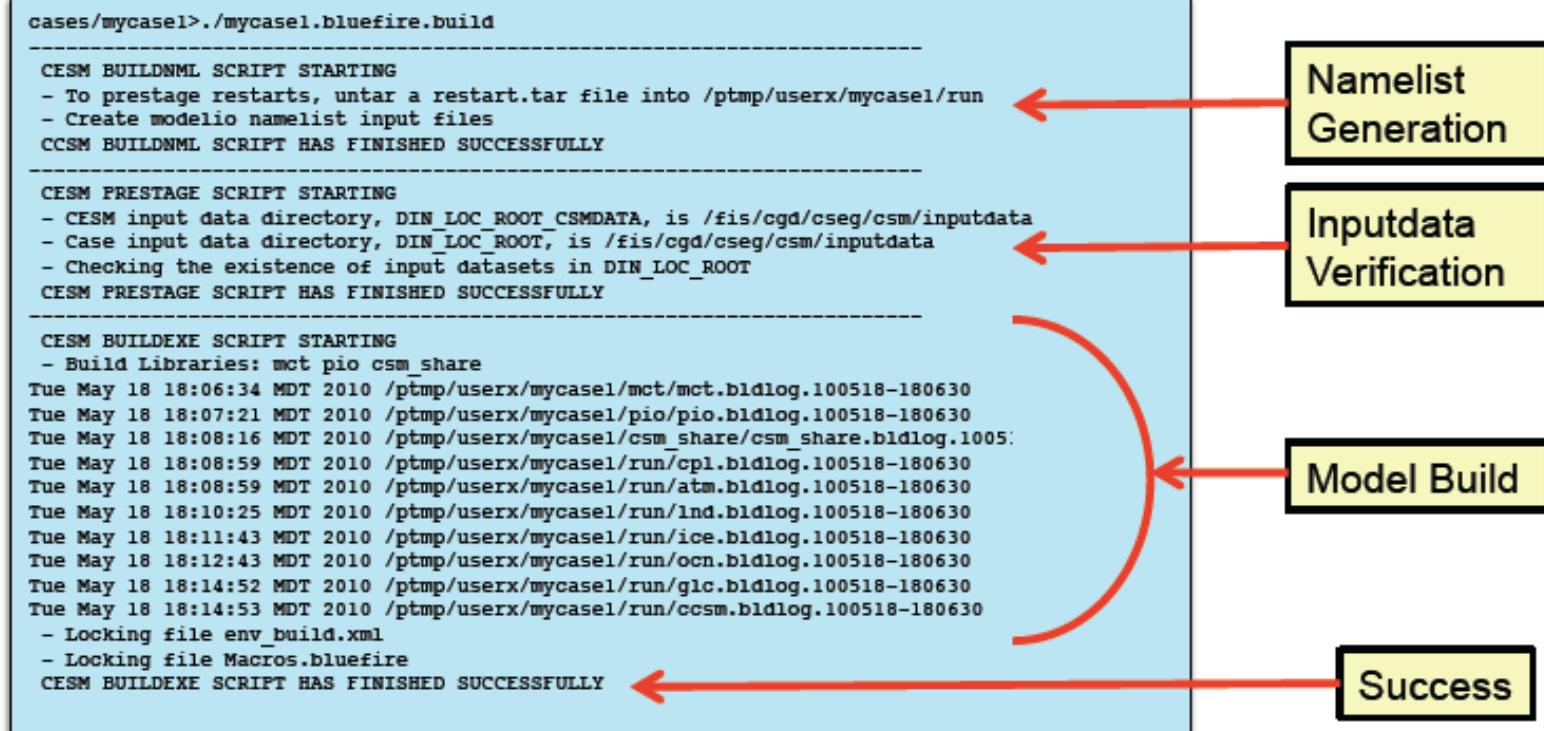
check_input_data

env_build.xml

.build script

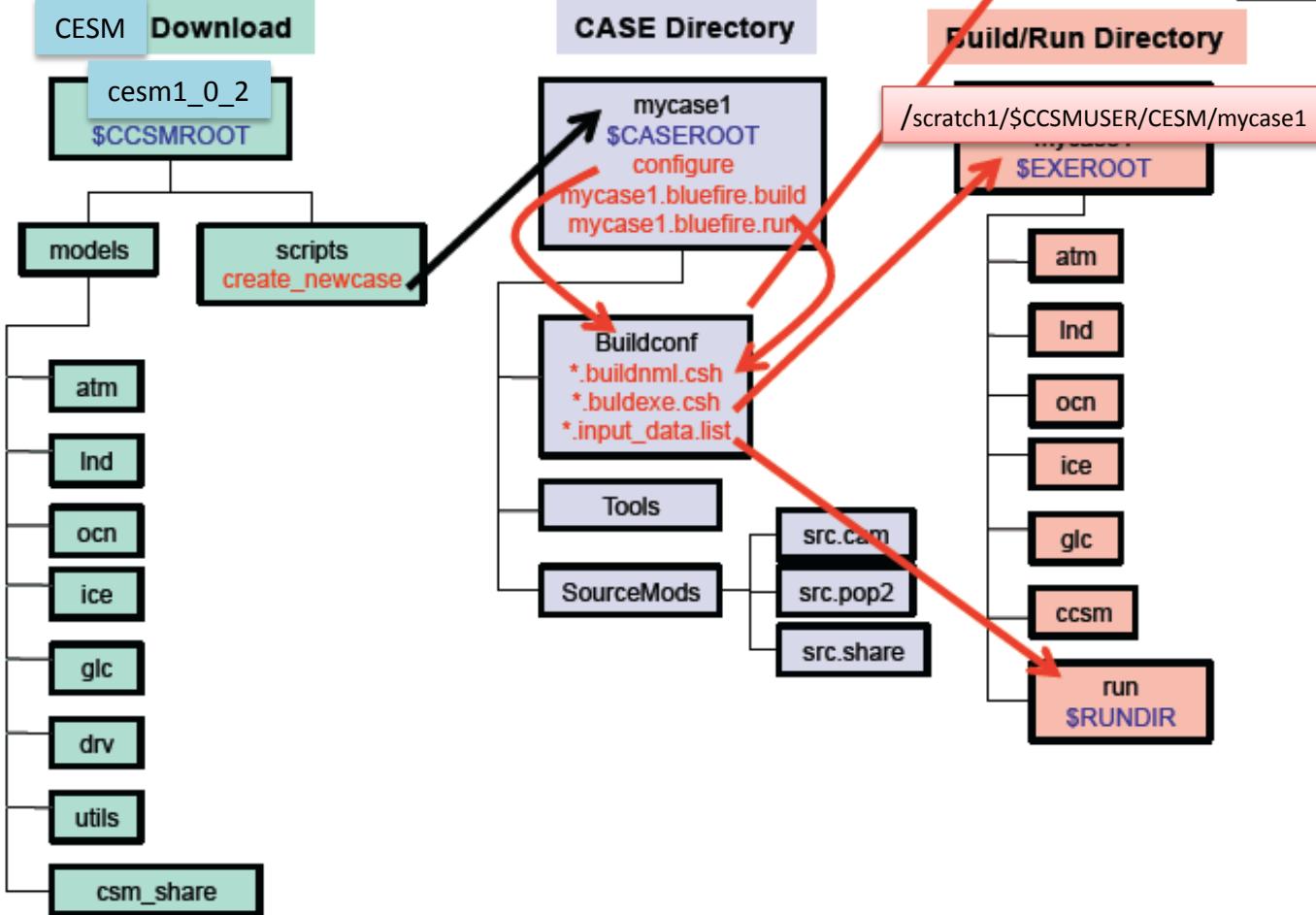
(3) Running the .build Script

- Checks for missing input data by running `check_input_data`
- If any inputdata is missing
 - the build will abort, but it will provide a list of missing files
 - you must run the script `check_input_data -export` and then re-build
 - `check_input_data -export` will use svn to put required data in the inputdata directory
- Generates input files (namelist) by running the `buildnml.csh` scripts for each component
- Builds the component model libraries and then final executable by running the `*.buildexe.csh` scripts for each component





(3) Overview of Directories (+ build)



(3) Your \$RUNDIR after running .build

```
cases/mycase1> source Tools/ccsm_getenv  
cases/mycase1> echo $RUNDIR  
> /ptmp/userx/mycase1/run  
cases/mycase1> ls -al $RUNDIR
```

```
cases/mycase1>ls -l $RUNDIR  
total 167552  
-rw-r--r-- 1 userx ncar 9960 May 18 18:10 atm.bldlog.100518-180630.gz  
-rw-r--r-- 1 userx ncar 2867 May 18 18:06 atm_in  
-rw-r--r-- 1 userx ncar 133 May 18 18:06 atm_modelio.nml  
-rw-r--r-- 1 userx ncar 1398 May 18 18:15 ccsm.bldlog.100518-180630.gz  
-rwxr-xr-x 1 userx ncar 84463482 May 18 18:15 ccsm.exe ← executable  
-rw-r--r-- 1 userx ncar 120 May 18 18:08 cpl.bldlog.100518-180630.gz  
-rw-r--r-- 1 userx ncar 133 May 18 18:06 cpl_modelio.nml  
-rw-r--r-- 1 userx ncar 50 May 18 18:06 drv_flds_in  
-rw-r--r-- 1 userx ncar 2545 May 18 18:06 drv_in  
-rw-r--r-- 1 userx ncar 589 May 18 18:14 glc.bldlog.100518-180630.gz  
-rw-r--r-- 1 userx ncar 133 May 18 18:06 glc_modelio.nml  
-rw-r--r-- 1 userx ncar 2569 May 18 18:12 ice.bldlog.100518-180630.gz ← .bld.log files  
-rw-r--r-- 1 userx ncar 3279 May 18 18:06 ice_in  
-rw-r--r-- 1 userx ncar 133 May 18 18:06 ice_modelio.nml  
-rw-r--r-- 1 userx ncar 4591 May 18 18:11 lnd.bldlog.100518-180630.gz  
-rw-r--r-- 1 userx ncar 1918 May 18 18:06 lnd_in ← namelist files  
-rw-r--r-- 1 userx ncar 133 May 18 18:06 lnd_modelio.nml  
-rw-r--r-- 1 userx ncar 3668 May 18 18:14 ocn.bldlog.100518-180630.gz  
-rw-r--r-- 1 userx ncar 133 May 18 18:06 ocn_modelio.nml  
-rw-r--r-- 1 userx ncar 14976 May 18 18:06 pop2_in  
-rw-r--r-- 1 userx ncar 1882 May 18 18:06 seq_maps.rc
```

Next Step In the Basic Work Flow

- **One-Time Setup Steps**
 - (A)Registration and Download**
 - (B)Create an Input Data Root Directory**
 - (C) Porting**
- **Creating & Running a Case**
 - (1) Create a New Case**
 - (2) Configure the Case**
 - (3) Build the Executable**
 - (4)Run the Model: Initial Run and Output Data Flow**
 - (5) Run the Model: Continuation Runs**

Mapache Quickstart

```
# set up the cesm environment on mapache
source /usr/projects/cesm/scripts/mapache/pgi/cesm_env_script

# go to root directory of cesm code and cd to scripts directory
cd cesm1_0_2/scripts

# (1) create a new case
./create_newcase -case mycase.1850 -res T31_g37 -compset B_1850 -mach mapache

# (2) configure the case
cd mycase.1850
./configure --case

# (3) build the case
./mycase.1850.mapache.build

# (4) submit the case
./mycase.1850.mapache.submit

#check job status
showq -w user=username
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# (5) when the run finishes, change the length from a continuation run to a month-long
#     run and continue the simulation, then check job status as above (from mycase dir)
xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE
./mycase.1850.mapache.submit

# (6) view job output after job has completed
source Tools/ccsm_getenv

cd $DOUT_S_ROOT/atm/hist
ncview mycase.1850.cam2.h1.0001-01-01-00000.nc
```



(4) Running the Model: an Initial Run

- Step (4) is to do an initial run of the model
- Initial runs are usually short (default is 5-days) used to verify that model is running correctly
- May want to edit env_run.xml file before running
 - Such as to specifying length of run
- May want to modify namelist settings before running
 - Via env_run.xml variables
 - Directly in the Buildconf/*.buildnml.csh files
- The run script
 - Generates the namelist files in \$RUNDIR (again)
 - Verifies existence of input datasets (again)
 - DOES NOT build (or re-build) the executable

```
cases/mycase1> bsub < mycase1.bluefire.run
```

```
cases/mycase1>bsub < mycase1.bluefire.run
Job <40597> is submitted to queue <regular>.
```

```
cases/mycase1> bjobs
```

cases/mycase1>bjobs							
JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
40597	userx	PEND	regular	bell05en		mycase1	May 18 18:30



(4) Output in Your CASE Directory

A job completed successfully if "SUCCESSFUL TERMINATION OF CPL7-CCSM" appears near end of the cpl.log file

```
cases/mycasel>ls -l
total 512
drwxr-xr-x  6 userx  ncar          8192 May 18 18:32 Buildconf
drwxr-xr-x  2 userx  ncar          8192 May 18 18:06 CaseDocs
drwxr-xr-x  2 userx  ncar          8192 May 18 18:15 LockedFiles
-rw-r--r--  1 userx  ncar         10687 May 13 14:32 Macros.bluefire
drwxr-xr-x  2 userx  ncar          8192 May 13 14:32 README
-rw-r--r--  1 userx  ncar             66 May 13 14:32 README.case
drwxr-xr-x  9 userx  ncar          8192 May 13 14:32 SourceMods
drwxr-xr-x  4 userx  ncar          8192 May 13 14:32 Tools
-rw-r--r--  1 userx  ncar         9330 May 12 11:33 check_input_data
-rw-r--r--  1 userx  ncar        10092 May 12 11:33 configure
-rw-r--r--  1 userx  ncar         3085 May 12 11:33 create_production_test
-rw-r--r--  1 userx  ncar         4475 May 18 18:32 env_build.xml
-rw-r--r--  1 userx  ncar         5635 May 13 14:32 env_case.xml
-rw-r--r--  1 userx  ncar         7029 May 13 14:32 env_conf.xml
-rw-r--r--  1 userx  ncar             614 May 18 18:37 env_derived
-rw-r--r--  1 userx  ncar         5916 May 13 17:12 env_mach_pes.xml
-rw-r--r--  1 userx  ncar         2199 May 13 14:32 env_mach_specific
-rw-r--r--  1 userx  ncar         10466 May 13 14:32 env_run.xml
drwxr-xr-x  3 userx  ncar          8192 May 18 18:37 logs
-rw-r--r--  1 userx  ncar             0 May 18 17:55 ls
-rw-r--r--  1 userx  ncar         270 May 18 18:37 poe.stderr.40597
-rw-r--r--  1 userx  ncar         2013 May 18 18:37 poe.stdout.40597
-rw-r--r--  1 userx  ncar             574 May 13 17:12 mycasel.bluefire.build
-rw-r--r--  1 userx  ncar             836 May 13 17:12 mycasel.bluefire.clean_build
-rw-r--r--  1 userx  ncar             802 May 13 17:12 mycasel.bluefire.l_archive
-rw-r--r--  1 userx  ncar             3938 May 13 17:12 mycasel.bluefire.run
drwxr-xr-x  2 userx  ncar          8192 May 18 18:37 timing
-rw-r--r--  1 userx  ncar         10388 May 12 11:33 xmlchange
cases/mycasel>ls -l logs
total 272
-rw-r--r--  1 userx  ncar         29882 May 18 18:37 atm.log.100518-183212.gz
drwxr-xr-x  2 userx  ncar          8192 May 18 18:15 bld
-rw-r--r--  1 userx  ncar         19115 May 18 18:37 ccm.log.100518-183212.gz
-rw-r--r--  1 userx  ncar         4998 May 18 18:37 cpl.log.100518-183212.gz
-rw-r--r--  1 userx  ncar         18732 May 18 18:37 ice.log.100518-183212.gz
-rw-r--r--  1 userx  ncar         9384 May 18 18:37 lnd.log.100518-183212.gz
-rw-r--r--  1 userx  ncar         18534 May 18 18:37 ocn.log.100518-183212.gz
cases/mycasel>ls -l timing
total 32
-rw-r--r--  1 userx  ncar         6204 May 18 18:37 ccm_timing.mycasel.100518-183212
-rw-r--r--  1 userx  ncar         3711 May 18 18:37 ccm_timing_summary.100518-183212.gz
```

Copies of the
Current Namelist
Input Files

stdout/err

Log Files

Timing Files



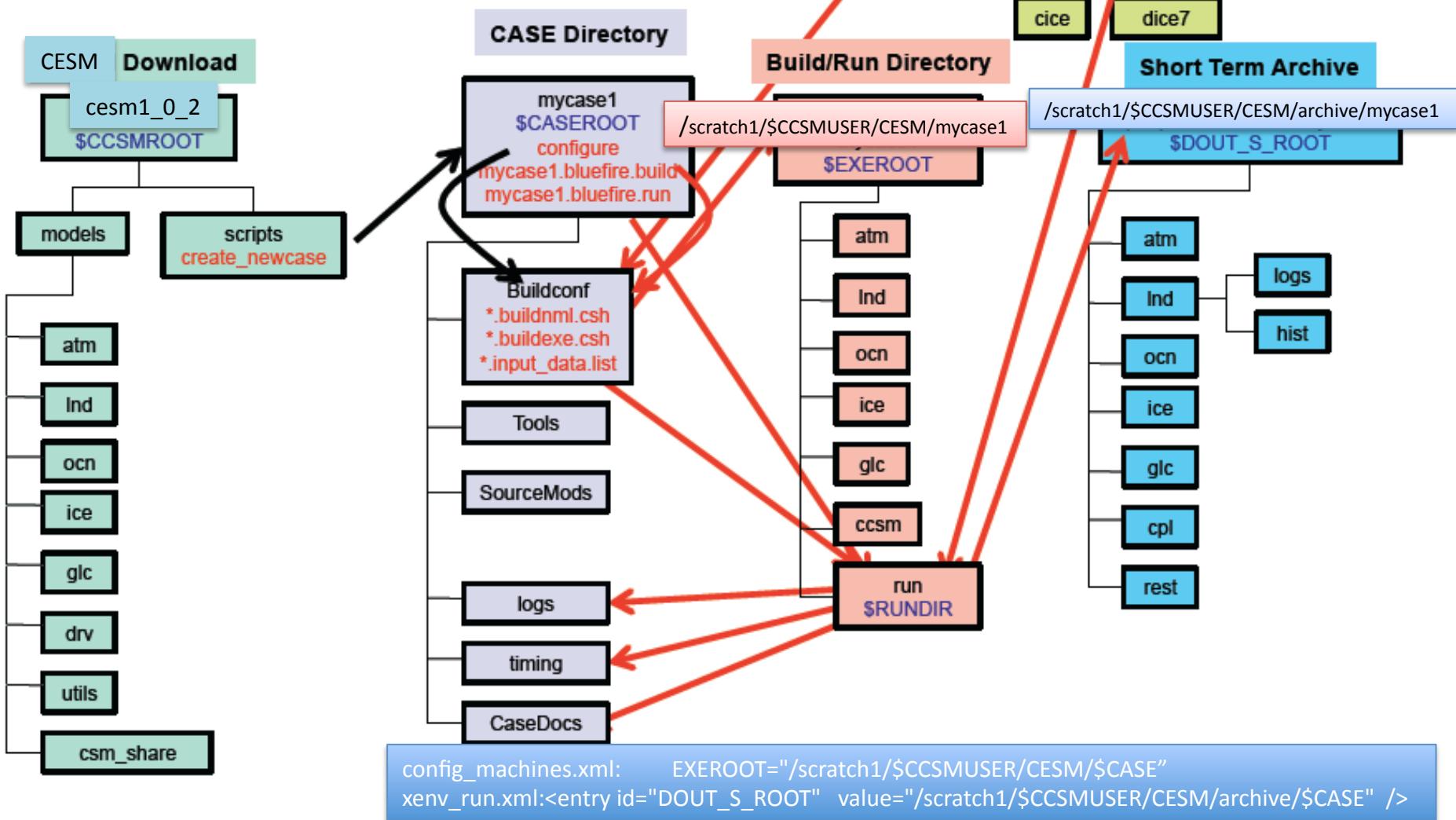
(4) Output in Short Term Archiving Directory

- Output data is originally created in **\$RUNDIR**
- When the run ends, output data is moved into a short term archiving directory, **\$DOUT_S_ROOT**
- Why?
 - Cleans up the \$RUNDIR directory
 - Migrates output data away from a possibly volatile \$RUNDIR
 - Gathers data for the long term archive script which can then save the data to a permanent long-term storage area (e.g. HPSS)

```
cases/mycasel>echo $DOUT_S_ROOT
/ptmp/usrxx/archive/mycasel
cases/mycasel>ls -l $DOUT_S_ROOT
total 1024
drwxr-xr-x 4 userx ncar 65536 May 18 18:37 atm
drwxr-xr-x 4 userx ncar 65536 May 18 18:37 cpl
drwxr-xr-x 4 userx ncar 65536 May 18 18:37 dart
drwxr-xr-x 3 userx ncar 65536 May 18 18:37 glc
drwxr-xr-x 4 userx ncar 65536 May 18 18:37 ice
drwxr-xr-x 4 userx ncar 65536 May 18 18:37 lnd
drwxr-xr-x 4 userx ncar 65536 May 18 18:37 ocn
drwxr-xr-x 3 userx ncar 65536 May 18 18:37 rest
cases/mycasel>ls -l $DOUT_S_ROOT/cpl
total 256
drwxr-xr-x 2 userx ncar 65536 May 18 18:37 hist
drwxr-xr-x 2 userx ncar 65536 May 18 18:37 logs
cases/mycasel>ls -l $DOUT_S_ROOT/cpl/logs/
total 256
-rw-r--r-- 1 userx ncar 19115 May 18 18:37 ccsm.log.100518-183212.gz
-rw-r--r-- 1 userx ncar 4998 May 18 18:37 cpl.log.100518-183212.gz
cases/mycasel>ls -l $DOUT_S_ROOT/ocn/hist
total 436608
-rw-r--r-- 1 userx ncar 3 May 18 18:32 mycasel.pop.dd.0001-01-02-00000
-rw-r--r-- 1 userx ncar 2787 May 18 18:36 mycasel.pop.do.0001-01-02-00000
-rw-r--r-- 1 userx ncar 3 May 18 18:32 mycasel.pop.dt.0001-01-02-00000
-rw-r--r-- 1 userx ncar 1183 May 18 18:36 mycasel.pop.dv.0001-01-02-00000
-rw-r--r-- 1 userx ncar 27046596 May 18 18:36 mycasel.pop.h.nday1.0001-01-02.nc
-rw-r--r-- 1 userx ncar 78164092 May 18 18:33 mycasel.pop.h.once.nc
-rw-r--r-- 1 userx ncar 117965260 May 18 18:32 mycasel.pop.hv.nc
```



(4) Overview of Directories (+ run + short term archive)



Next Step In the Basic Work Flow

- **One-Time Setup Steps**
 - (A)Registration and Download
 - (B)Create an Input Data Root Directory
 - (C) Porting
- **Creating & Running a Case**
 - (1)Create a New Case
 - (2)Configure the Case
 - (3) Build the Executable
 - (4)Running the Model: Initial Run and Output Data
 - (5) Running the Model: Continuation Runs**

Mapache Quickstart

```
# set up the cesm environment on mapache
source /usr/projects/cesm/scripts/mapache/pgi/cesm_env_script

# go to root directory of cesm code and cd to scripts directory
cd cesm1_0_2/scripts

# (1) create a new case
./create_newcase -case mycase.1850 -res T31_g37 -compset B_1850 -mach mapache

# (2) configure the case
cd mycase.1850
./configure --case

# (3) build the case
./mycase.1850.mapache.build

# (4) submit the case
./mycase.1850.mapache.submit

#check job status
showq -w user=username
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# (5) when the run finishes, change the length from a continuation run to a month-long
#     run and continue the simulation, then check job status as above (from mycase dir)
xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE
./mycase.1850.mapache.submit

# (6) view job output after job has completed
source Tools/ccsm_getenv

cd $DOUT_S_ROOT/atm/hist
ncview mycase.1850.cam2.h1.0001-01-01-00000.nc
```

(5) Running the Model: Continuation Runs

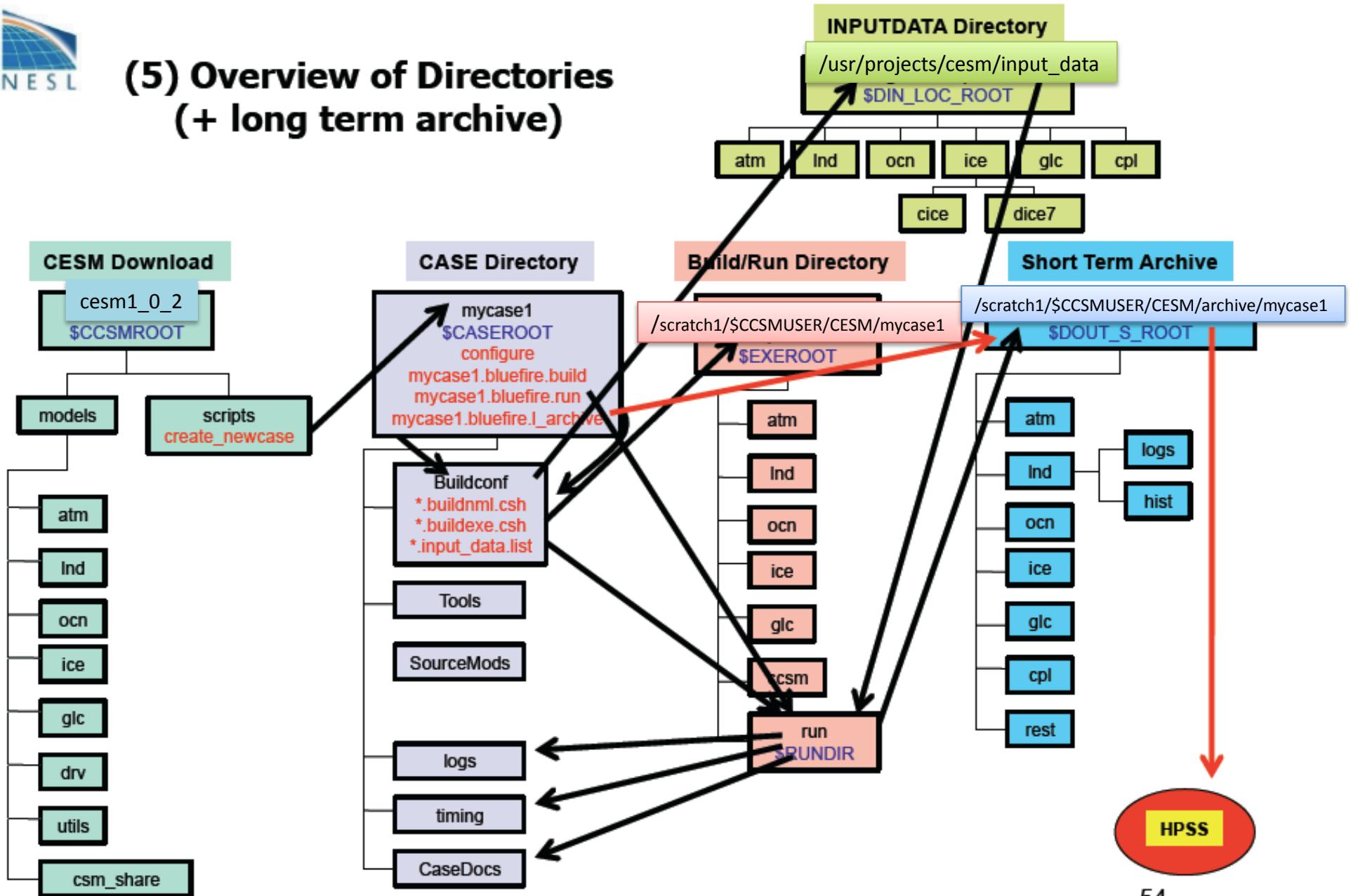
- You should start with a short initial run, described in step (4)
- Carefully examine the output to verify that the run is doing what you want it to
 - You might rerun the initial run several times to fix problems
- If the initial run looks good...step (5) is a continuation run
 - Change CONTINUE_RUN to TRUE in env_run.xml
 - Probably change STOP_OPTION to run the model longer
 - May want to turn on auto-resubmit option in env_run.xml (RESUBMIT)
 - May want to turn on “long term archiving” in env_run.xml (DOUT_L_MS)

(5) Long Term Archiving

- Why?
 - Migrates output data away from a possibly volatile *\$DOUT_S_ROOT* into a permanent long-term storage area
 - Long term archiving script moves data conveniently and in parallel
- To turn on short term archiving (default is on)
 - Set DOUT_S to TRUE in env_run.xml
- To turn on long term archiving (default is off)
 - Set DOUT_L_MS to TRUE in env_run.xml;
 - Causes run script to automatically submit a long term archiver job ([.l_archive](#)) at the end of every successful run.
 - Long term archiver
 - moves data from the short term archive directory to a long term archiving system (e.g. HPSS) - if one exists
 - runs in batch on one processor
 - can run in parallel with a production job
 - will not interfere with a production job or vice versa.

Long term archiving via script settings with CESM may not be feasible, since archiving in the turquoise requires username/cryptocard password.

(5) Overview of Directories (+ long term archive)



What's in the output files? Use utility ncdump to dump file metadata (or the whole file):

```
mp-fe2.lanl.gov/~scratch1/rauscher/CESM/archive/mycase.1850/atm/hist % ncdump -h mycase.1850.cam2.h0.0001-01.nc
```

```
netcdf mycase.1850.cam2.h0.0001-01 {
```

```
dimensions:
```

```
lat = 48 ;  
lon = 96 ;  
lev = 26 ;  
ilev = 27 ;  
isccp_prs = 7 ;  
isccp_tau = 7 ;  
isccp_prstau = 49 ;  
time = UNLIMITED ; // (1 currently)  
tbnd = 2 ;  
chars = 8 ;
```

```
variables:
```

```
double P0 ;  
P0:long_name = "reference pressure" ;  
P0:units = "Pa" ;  
double lat(lat) ;  
lat:long_name = "latitude" ;  
lat:units = "degrees_north" ;  
double lon(lon) ;  
lon:long_name = "longitude" ;  
lon:units = "degrees_east" ;  
double lev(lev) ;  
lev:long_name = "hybrid level at midpoints (1000*(A+B))" ;  
lev:units = "level" ;  
lev:positive = "down" ;  
lev:standard_name = "atmosphere_hybrid_sigma_pressure_coordinate" ;  
lev:formula_terms = "a: hyam b: hybm p0: P0 ps: PS" ;  
nbdate:long_name = "base date (YYYYMMDD)" ;
```

```
CLDLOW:Sampling_Sequence = "rad_lwsw" ;
```

```
CLDLOW:units = "fraction" ;
```

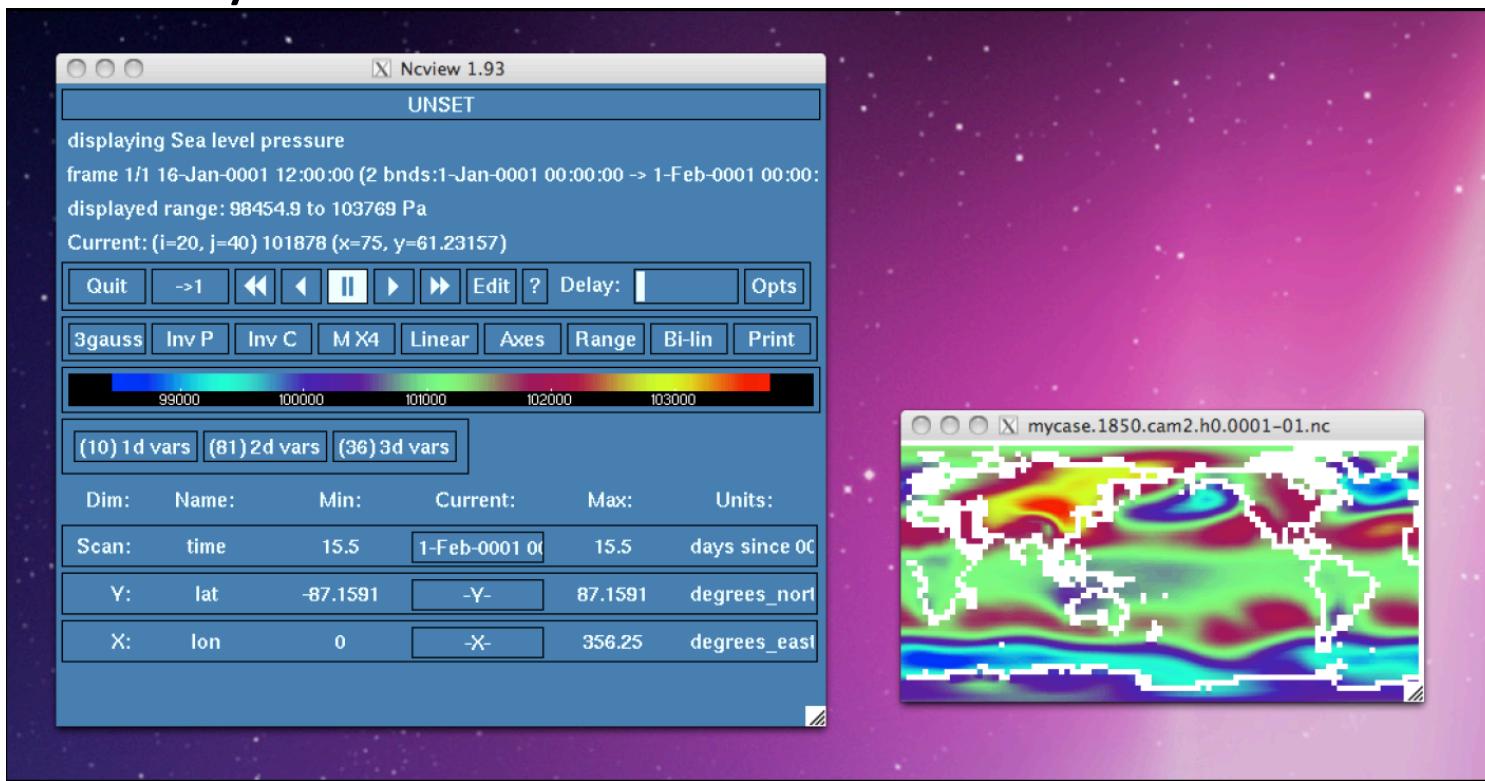
```
CLDLOW:long_name = "Vertically-integrated low cloud" ;
```

```
CLDLOW:cell_methods = "time: mean" ;
```

Visualizing Output

- You can use any software that can visualize netCDF data
- Here we will use ncview to take a quick look at our data from the 1850 simulation

ncview mycase.1850.cam2.h0.0001-01.nc



NCO: netCDF operators

NCO is a suite of programs (run on command line) designed to perform certain operations (differencing, subsetting, ensemble averaging...) on netCDF files.

It's incredibly useful and powerful, especially when used in shell scripts.

NCO homepage <http://nco.sourceforge.net>

```
ncdiff file1 file2 file1-file2.nc
```

```
ncks -v PSL mycase.1850.cam2.h0.0001-01.nc psl.nc
```

For a more comprehensive introduction to post-processing and visualization, take a look at Practical Session 3: Intro: Diagnostics and Output by Adam Phillips (.pdf and video available) on the CESM tutorial page.



More Information/Getting Help

- **Model User Guides (please provide feedback)**
 - <http://www.cesm.ucar.edu/models/cesm1.0/>
 - CESM Users Guide and Web-Browseable code reference
 - CAM, CLM, POP2, CICE, Data Model and CPL7 Users Guides
- **CESM Bulletin Board**
 - <http://bb.cgd.ucar.edu/>
 - Facilitate communication among the community
 - Ask questions, look for answers
 - Many different topics
- **CESM Release Page Notes**
 - <http://www.cesm.ucar.edu/models/cesm1.0/tags/>
 - Notes significant bugs or issues as they are identified
- **Model output is available on the Earth System Grid**
 - <http://www.earthsystemgrid.org>
- **Getting Help - email**
 - cesm-help@cgd.ucar.edu
 - Questions will be answered as resources are available

(A) Download the Source Code

- Code and input datasets are in a subversion repository (get subversion at <http://subversion.apache.org/>)
- Do NOT download any input data

> **svn list --username guestuser https://svn-ccsm-release.cgd.ucar.edu/model_versions**

```
>svn list --username guestuser https://svn-ccsm-release.cgd.ucar.edu/model_versions
Error validating server certificate for 'https://svn-ccsm-release.cgd.ucar.edu:443':
- The certificate is not issued by a trusted authority. Use the
  fingerprint to validate the certificate manually!
- The certificate hostname does not match.
- The certificate has expired.
Certificate information:
- Hostname: localhost.localdomain
- Valid: from Wed, 20 Feb 2008 23:32:25 GMT until Thu, 19 Feb 2009 23:32:25 GMT
- Issuer: SomeOrganizationalUnit, SomeOrganization, SomeCity, SomeState, --
- Fingerprint: 86:01:bb:a4:4a:e8:4d:8b:e1:f1:01:dc:60:b9:96:22:67:a4:49:ff
(R)eject, accept (t)emporarily or accept (p)ermanently? p
Authentication realm: <https://svn-ccsm-release.cgd.ucar.edu:443> cccsm:release
Password for 'guestuser': *****
ccsm4_0/
cesm1_0/
/
```

login

password

> **svn co --username guestuser https://svn-ccsm-release.cgd.ucar.edu/models_versions/cesm1_0**

```
>svn co --username guestuser https://svn-ccsm-release.cgd.ucar.edu/model_versions/cesm1_0
A   cesm1_0/models
A   cesm1_0/models/dead_share
A   cesm1_0/models/dead_share/dead_data_mod.F90
A   cesm1_0/models/dead_share/dead_mod.F90
A   cesm1_0/models/dead_share/dead_mct_mod.F90
A   cesm1_0/models/ocn
A   cesm1_0/models/ocn/pop2
...
```



(3) Modifying Source Code

- Code modified in the models directory will apply to all new cases created – PROBABLY A BAD IDEA
- Modified code in the CASE SourceMods subdirectory applies to that case only
- Files in the SourceMods/ must be in proper subdirectory, eg. pop2 code must be in src.pop2

